

INFO-H-511: Web Services

Lab 5 - Presenting Web Services

Lecturer: Stijn Vansummeren
Teaching Assistant: Stefan Eppe
<http://cs.ulb.ac.be/public/teaching/infoh511>
2013–2014

Part I: Setup

In this lab, we will implement a web application that uses a RESTful address book service. The source code of the web service itself is included in the lab repository.

To install this code, you will need to copy the “contacts” directory provided in the repository to your home directory¹. After importing the project residing in the “RESTContactManager” directory into Eclipse, run the main entry point from the `be.ac.ulb.code.wit.ResourcesServer` class. The console should display

Jersey app started with WADL available at <http://0.0.0.0:8080/application.wadl>

Opening a browser on <http://0.0.0.0:8080/> should display an address book web application.

Part II: JSON

Our address book supports a JSON representation of its data, to communicate with the web application. It supports the following operations:

| URL | Method | Semantics |
|---|----------------|--|
| <code>/contacts</code> | GET | Retrieve a list of links to each contacts |
| | POST | Create a new contact and add it to the list |
| <code>/contacts/{id}</code> | GET/PUT/DELETE | Retrieve/modify/remove the contact corresponding to <code>id</code> |
| <code>/groups</code> | GET | Retrieve a list of links to each group of contacts |
| | POST | Create a new group of contacts |
| <code>/groups/{name}</code> | GET/PUT/DELETE | Retrieve/rename/remove the named group of contacts. The representation of a group contains a link to its list of contacts. |
| <code>/groups/{name}/contacts</code> | GET | Retrieve the contacts of the specified group |
| | POST | Add a contact to the group. If the posted document does not contain an existing <code>id</code> , a new contact will be created. |
| <code>/groups/{name}/contacts/{id}</code> | GET/PUT | Retrieve/modify the specified contact |
| | DELETE | Remove the contact <i>from the group</i> . |

¹Alternately, you can update the `ServiceConfiguration.java` source file to provide the right directory.

Exercise 5.1

Devise a JSON representation of the different resources. For this purpose, provide an example JSON document for each of the following:

- A contact (consisting of a first name, last name, nickname, email address, and a list of groups)
- A group of contacts (consisting of a name)
- The list of all contacts
- The list of contacts of a group

Exercise 5.2

Using `curl`, check how the provided implementation represents its information. For this purpose, you can invoke `curl` as:

```
curl -H "Accept: application/json" http://localhost:8080/contacts/2
```

Part III: An Address Book Web Application

In a REST setting, web applications often offer an HTML representation of the data for users to consume. Interaction with the application is made possible by linking various pages and forms. In this setting, each time the user performs an action, a new page is loaded.

While this kind of interaction is flexible and is the only interaction supported by older web browsers, many modern web applications need to update content without having to load a new page. AJAX (Asynchronous JavaScript and XML) is often used for this purpose. Essentially, it is a technique that allows sending and HTTP request programmatically from JavaScript code. The data thereby retrieved is then parsed and acted upon by the application.

Exercise 5.3

Our address book web application relies on AJAX and JSON to update its display ². Open the application in a web browser and do the following:

- A web development tool such as Firebug and browse the application. Identify the json resources that are loaded, and their purpose.
- Open the `contacts.js` file. The `updateGroups` function loads the different groups. Explain how it proceeds.

Exercise 5.4

The contact removal feature has not been implemented yet. Using the `unsetGroup` function as an example, implement the `removeContact` function.

Exercise 5.5

In the contact display, we want to retrieve a profile picture from `http://en.gravatar.com/`. Using `https://en.gravatar.com/site/implement/images/` as a reference, update the `showContactLink` function to add an `emailHash` field to the `data`. You can generate a hash by calling the `md5` function. Update the `contact.mustache` template to refer to the right url.

²Real-world applications should always provide alternatives to AJAX, to support older browsers or browsers where scripts are disabled.